Hierarchical Routes in ArcGIS® Network Analyst

An ESRI® White Paper • July 2005

ESRI 380 New York St., Redlands, CA 92373-8100, USA • TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB www.esri.com

Hierarchical Routes in ArcGIS Network Analyst

An ESRI White Paper

Contents **Page**

Introduction 1

Hierarchical versus Exact Routes 1

Hierarchical Algorithm Concepts 3
Hierarchical Classification of a Network 3
Overview of the Hierarchical Algorithm 4
Turn Considerations 6
Barrier Considerations 7
n by m Best Routes 8

Network Modeling Concepts 10
Creating Hierarchical Levels 10
Modeling Considerations 11
StreetMap Network Considerations 13

Using Hierarchy in Network Analysis 14
Best Route Algorithm Parameters 14
Importance of Parameters 16
Impedance 16
Restriction Attributes, Accumulate Attributes, and U-Turn Policy 18
Hierarchy Options 18
Assess Solution Quality and Performance 21
Diagnostic Problems 22

Summary 23

Contents **Page**

Appendixes

Appendix A—Details of the Hierarchical Route Algorithm 25
Appendix B—Examples of Hierarchy Modeling 27

Hierarchical Routes in ArcGIS Network Analyst

Introduction

Hierarchical algorithms are useful for finding a route between an origin and a destination on large networks such as nationwide networks. The overall objective is to sacrifice some accuracy for speed. In cases such as street networks where there is a natural hierarchy, the trade-off is well worth it. Hierarchical algorithms have been a major contributor to the success of various ESRI® routing products such as NetEngine™, ArcLogistics™ Route, the ArcGIS® StreetMap™ extension, and ArcIMS® Route Server.

The ArcGIS Network Analyst extension includes a hierarchical algorithm that has been designed to support the rich modeling capabilities of the network dataset available in ArcGIS 9.1 and to support StreetMap data, which can be used for routing. Furthermore, it has been designed to be the core engine for three different Network Analyst functions, namely Route, Closest Facility, and Origin-Destination (OD) Cost Matrix. In ArcGIS, network analyses using the hierarchical algorithm can be performed in ArcMap™, using a Geoprocessing tool, in a custom stand-alone application developed with ArcGIS Engine, or via a Web application developed using ArcGIS Server.

The goal of this paper is to provide useful and practical information for desktop users and developers building desktop or server applications who would like to understand how to use this hierarchical algorithm effectively.

This paper assumes that the reader is familiar with using ArcGIS and has a basic understanding of the network dataset and ArcGIS Network Analyst.

This paper discusses a comparison between exact and hierarchical best route algorithms and introduces hierarchical solver concepts. It then provides information on modeling hierarchical networks with typical data and commercial data. Tips on how to efficiently use the hierarchy settings during network analysis with regard to performance and solution quality as well as conclusions are presented.

Hierarchical versus Exact Routes

Finding the best route in ArcGIS Network Analyst consists of computing the quickest or shortest route on the network, depending on the impedance chosen. In this section, two best route algorithms implemented in ArcGIS Network Analyst are compared: the exact best route algorithm and the hierarchical best route algorithm. Both exact and hierarchical best route algorithms

- Minimize one impedance (e.g., drive time or length).
- Find node-to-node, node-to-edge, edge-to-node, and edge-to-edge routes.

- Handle restrictions on edges and nodes (e.g., one-way restriction).
- Handle barriers to avoid restricted network elements (edges and nodes).
- Respect prohibited maneuvers (turn restriction).
- Add delay during maneuvers (turn penalty).
- Honor the curb approach specified for the origin and the destination (e.g., depart from the right side of street).

The list below analyzes the main differences between exact and hierarchical best route algorithms.

- An exact best route algorithm[1] computes the optimal route between an origin and a destination, which may require traversing the entire network. In contrast, a hierarchical best route algorithm may compute a suboptimal route, longer in terms of impedance, but the route may be more realistic. The basic assumption on which hierarchy relies is that primary roads (e.g., highways) are faster than local roads (e.g., local streets). If users create a route that does not favor hierarchy, the total cost may be lower on this exact route than a hierarchical route. However, this may not be an accurate representation of reality. The exact route may not take into account delays such as wait times at traffic lights and stop signs. The hierarchical route favors primary roads based on the assumption that such delays are fewer on primary roads.
- The classical exact best route algorithm, well-known as the shortest-path algorithm, cannot support real-time queries on large networks. Sophisticated performance-enhancing techniques, such as heuristics and hierarchical algorithms, dramatically reduce the time needed to solve the problem.
- Driving directions, step by step instructions for navigating a route, are more concise and easier for drivers to follow when they are generated with a hierarchical best route algorithm. As stated earlier, a hierarchical algorithm favors a major road, which reduces the number of transitions between different road classes or road names. An exact best route algorithm may generate driving directions that contain too many instructions.
- Hierarchical routes may also be created when routing vehicles that have a preference for certain road classes. For example, when routing trucks, a hierarchy with a preference for primary roads can be used.
- Computation of all the routes between multiple origins and multiple destinations is required in many problems such as the traveling salesperson problem and the vehicle routing problem. These types of problems require computing n by m routes in an

---

[1] Dijskstra, E.W. "A note on two problems in connection with graphs." *Numero Math.* 1, 1959, 269–271.

efficient manner (as fast as possible with reasonable accuracy). An exact algorithm will compute all optimal routes but may consume extensive computing resources quickly. A hierarchical algorithm will sacrifice some accuracy for speed.

Hierarchical Algorithm Concepts

To provide an efficient hierarchical algorithm to users, a two-step procedure has been designed and implemented in Network Analyst.

1. A preprocessing step classifies a network into hierarchical levels.

2. During analysis, a hierarchical best route algorithm explores the preprocessed network to compute the best route between stops.

In this section, the preprocessing step is discussed, followed by an overview of the hierarchical algorithm used during analysis. Then, three advanced concepts are presented. The first one involves turns, the second one involves barriers, and the last one relates to n by m routes computation.

Hierarchical Classification of a Network

The preprocessing step classifies the edges of a given network into hierarchical levels; thus, it reduces the search space during the procedure of finding a route. The hierarchical classification is a spatial multilevel clustering that is performed when the network dataset is built. During the build process, a hierarchical attribute, specified by the user, is used to create up to three hierarchy levels of edge sources composed of

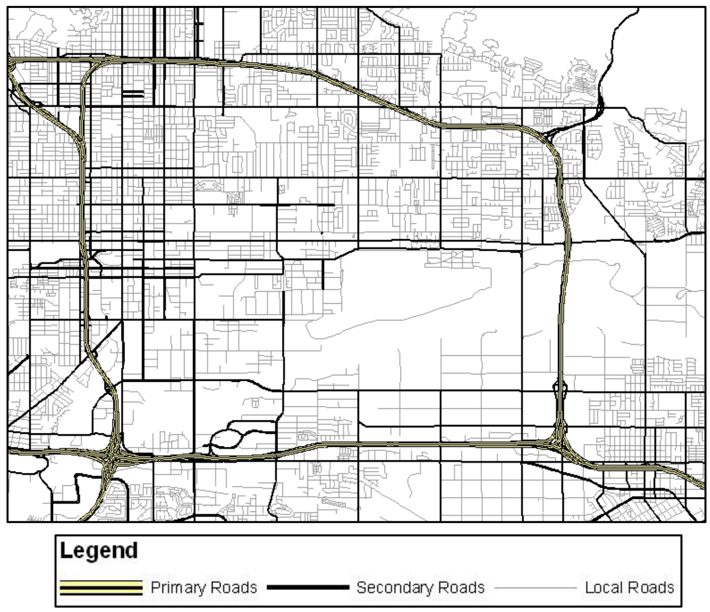
- Primary roads, which have the highest level (level 1)—for example, freeways and limited-access highways

- Secondary roads, which have the middle level (level 2)—for example, major roads and arterial roads

- Local roads, which have the lowest level (level 3)—for example, collectors and local streets

The figure below provides an example of a road network modeled using three levels of hierarchy in the Southern California area.

Figure 1
Example of Hierarchical Network



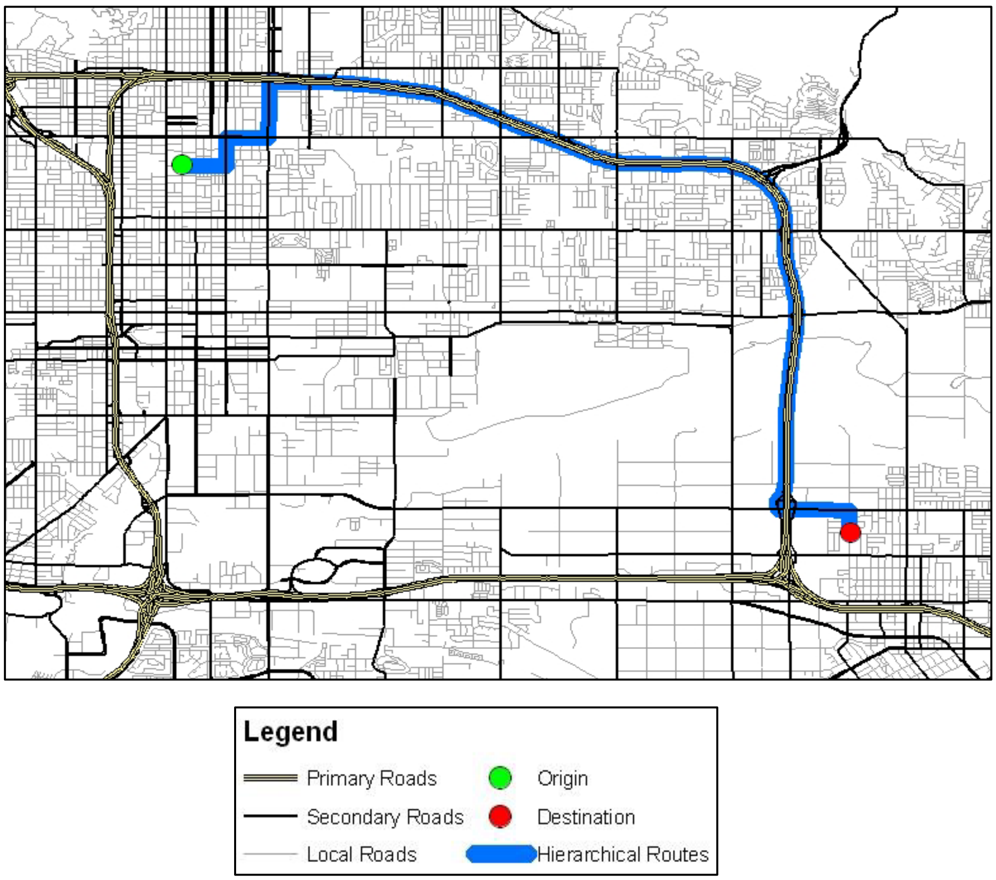
Overview of the Hierarchical Algorithm

Once the hierarchical network is created, a bidirectional hierarchical algorithm is used to compute the route between an origin and a destination during analysis. The overall objective of this technique is to minimize impedance while favoring a higher order of hierarchy. More precisely, the technique consists of

- Simultaneously searching from the origin and the destination until a specified number of connection points to the next higher level of hierarchy is found
- Exploring the highest level of hierarchy from the origin, disregarding all other levels
- Stopping the search when both segments of the route (one from the origin, and the other from the destination) meet

Figure 2 below presents an example of hierarchical route finding. Appendix A, Detail of the Hierarchical Route Algorithm, provides more explanation about the hierarchical best route algorithm implemented in the Network Analyst route solver.

Figure 2
Example of Hierarchical Route



Notes

- The approach described above is a heuristic search method that finds valid solutions but does not guarantee optimality as compared to the exact best route algorithm.
- This strategy assumes that the highest order of hierarchy is connected. If hierarchy level 1 is disconnected, the hierarchical algorithm will not descend to the lower levels when a dead end is encountered on level 1. In a worst-case scenario, it will not find a route.
- The network exploration from the origin and the destination is performed using a forward and a backward traversal trees search, respectively.
- The number of transitions to hierarchy parameters, detailed in the subsection Hierarchy Options, specifies the number of connection points to the next higher level

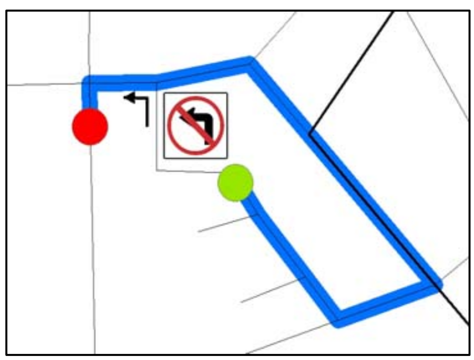
of hierarchy for both the origin and the destination. For example, in road networks, a connection point represents the ramp access from a boulevard to a freeway.

Turn Considerations

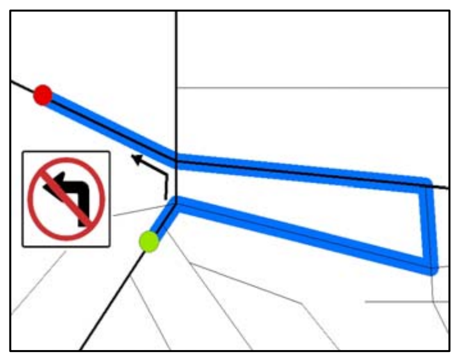
When growing the forward tree from the origin or the backward traversal tree from the destination, turn restrictions and penalties are taken into account. The turns may involve edges belonging to the same or different hierarchy levels.

Turns are handled by keeping additional information during the solve operation that can determine if the forward and backward traversal trees have met in a valid manner. When the forward and backward traversal trees meet at a common junction, the route through that junction might not be traversable due to a turn restriction.

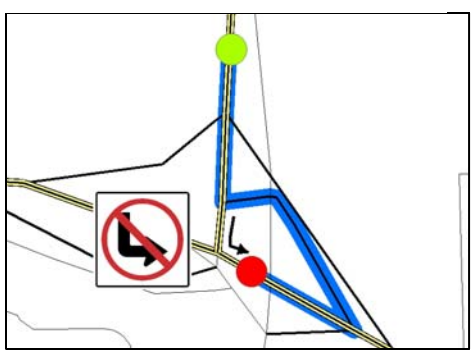
Figure 3
Examples of Hierarchical Routes Encountering Turn Restrictions



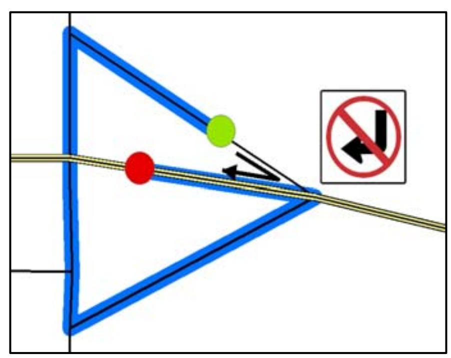
a. Turn restriction between two edges on level 3



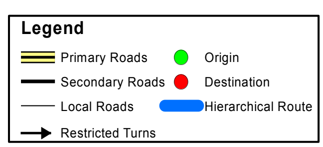
b. Turn restriction between two edges on level 2



c. Turn restriction between two edges on level 1



d. Turn restriction between one edge on level 2 and one edge on level 1



The first three (a, b, and c) show different routes involving turn restrictions between two edges belonging to the same hierarchy level. The last (d) represents a route where a turn restriction involves two edges belonging to two different levels of hierarchy.

J-9458

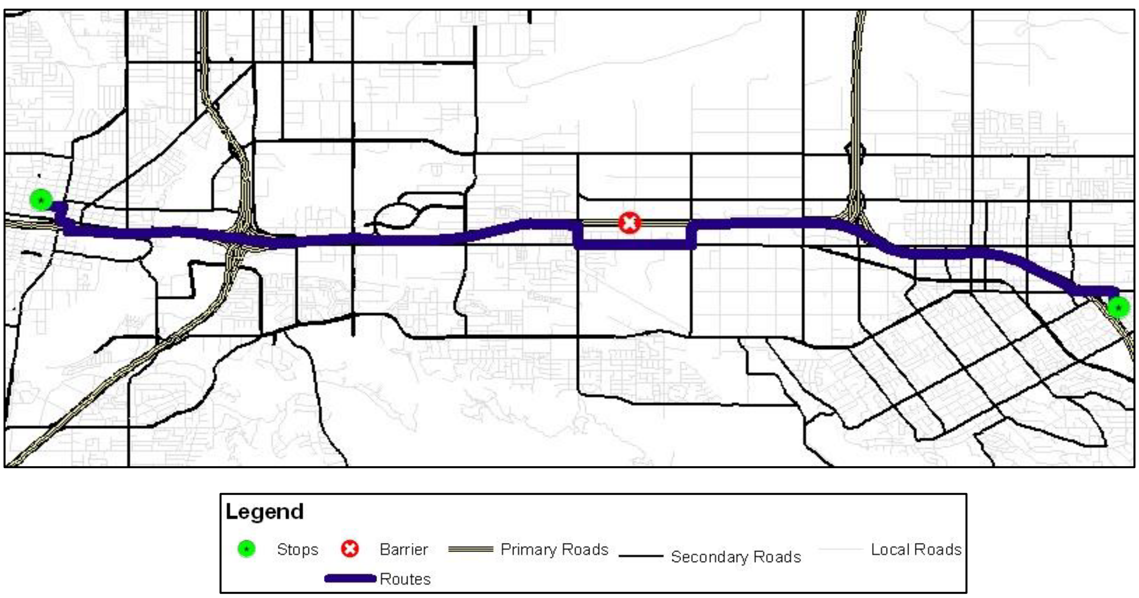
Barrier Considerations

The hierarchical algorithm can handle barriers located on network elements (edge or node). It also respects the barriers' properties: the edge can be restricted for both sides of the street or only the right or left side of the street.

When a barrier placed on an edge on the lowest hierarchy (level 3) is encountered during the search, the related network element is simply not considered in the forward and backward search.

For barriers on network elements in hierarchy levels 1 and 2, a special procedure has been implemented to handle them properly. When such a barrier is encountered, the route will descend to a lower hierarchy level, go around the barrier, then ascend back to the original hierarchy level. For example, when a barrier on an edge in level 1 is encountered, the hierarchy filter is relaxed and a larger subgraph is searched until a sufficient number of entries onto level 1 have been found. The hierarchy filter can then be tightened again.

Figure 4
Example of Hierarchical Route Descending to Avoid One Barrier Located on Hierarchy Level 1



Notes

- As the procedure explores more solutions locally around a barrier, the computation time may increase. This may be more significant when dealing with many barriers on a large network.

- The descent strategy for barriers on hierarchy levels 1 and 2 prevents an inferior route or none at all from being returned when the barrier disconnects hierarchy level 1.

n by m Best Routes

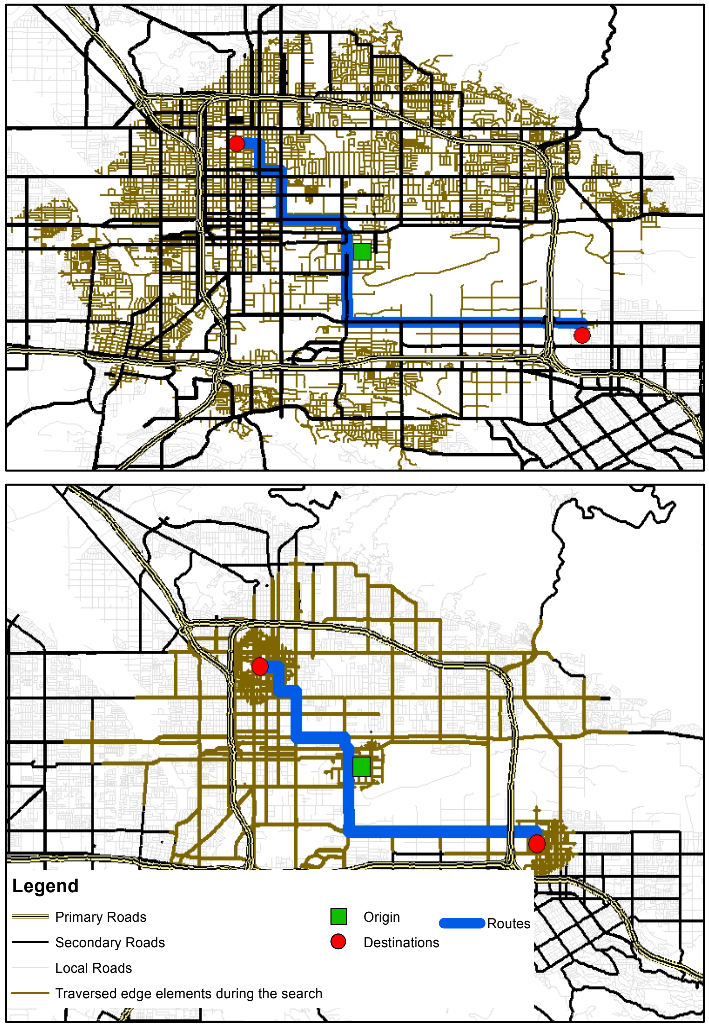
Solving an OD Cost Matrix or Closest Facility problem in Network Analyst consists of computing multiple routes from multiple n origins to multiple m destinations. It might require finding the k-closest destinations or finding routes to all destinations and ranking them in terms of cost. The n by m problem is decomposed into n 1 by m subproblems that are iteratively solved.

When solving the 1 by m best route problem using the exact best route algorithm, the forward traversal tree is simply grown from the origin until all m destinations are reached. Then, all routes are ranked based on the computed cost.

The 1 by m best route problem is solved differently when using the hierarchical best route algorithm. The hierarchical bidirectional algorithm presented in the subsection Overview of the Hierarchical Algorithm is used as the core algorithm for solving the 1 by m best route problem on hierarchical networks, with slight modification. The following specific operations are performed:

- The routes to reach the m destinations are computed using one forward traversal tree for the origin and m backward traversal trees from the destinations.
- When all destinations must be reached, all backward traversal trees are first computed.
- When only the k-closest destinations are needed, the backward traversal trees are computed as needed.

Figure 5
Example of n by m Exact and Hierarchical Best Routes



Examples of network edges traversed when computing multiple routes from one origin to two destinations using the exact best route algorithm (top) and the hierarchical best route algorithm (bottom). The exact best route grows the forward traversal tree from the origin until all destinations are reached. The hierarchical best route algorithm grows only the forward and backward traversal trees around the origin and all destinations until the highest hierarchy level is reached, then it traverses the highest hierarchy from the origin to compute all routes.

Notes

- When multiple origins are involved, all computed destination trees for the first origin are reused for the remaining origins.

- Reusing the computed backward traversal trees speeds up the solve operation. Effectively, all backward traversal trees computed for a given origin are held in memory, and only the forward traversal tree must be grown for each subsequent origin.

- The n by m hierarchical route algorithm solves to completion 1 by m problems on large networks (e.g., 1 origin and 1,000 destinations spread out on the entire road network of the United States). In such a case, the exact route solver may run out of memory and fail. Conversely, for small networks, the hierarchical solver may be too slow on 1 by m problems with many destinations, and the exact route solver should be used instead.

- Solving large n by m problems requires a lot of RAM; otherwise, the application will spend too much time swapping or, in a worst-case scenario, run out of memory. The amount of memory required depends on the size of the network and the problem instance. For example, on a Southern California shapefile-based network with 1.3 million edges, the following n by m problems can be solved to completion.

  - A 500 by 500 problem on a machine with 512 MB of RAM
  - A 1,000 by 1,000 problem on a machine with 1 GB of RAM
  - A 2,000 by 2,000 problem on a machine with 2 GB of RAM

Network Modeling Concepts

For most routing solutions, data quality is crucial to get good routes. Moreover, data processing is perhaps the most important task when building routing solutions for large networks. The data structure storing the connectivity and attribute information (e.g., cost and restriction) are often tied to the routing algorithm. This section provides information on modeling hierarchical networks in Network Analyst to get accurate routes meeting users' expectations.

Creating Hierarchical Levels

As mentioned in the previous section, the hierarchical levels for a given network are created during the network dataset build process. An efficient algorithm is used to build the network connectivity allowing navigation on the same hierarchy level or between hierarchy levels (e.g., ascending from the local roads up to the secondary roads). This algorithm also spatially clusters the network elements based on hierarchy groups. The details of this spatial multilevel clustering build algorithm are out of the scope of the present paper.

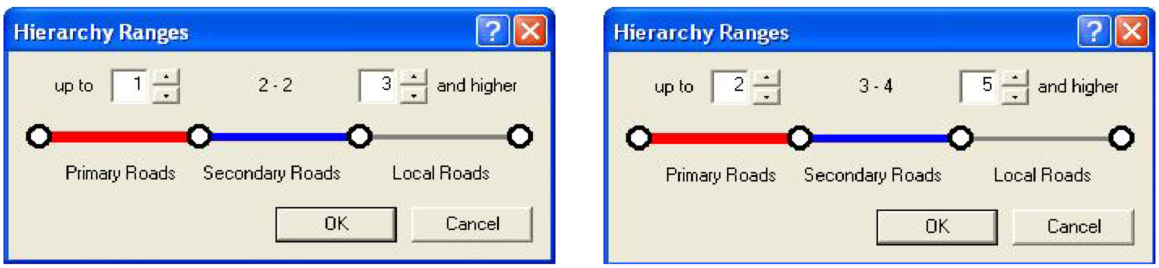
The hierarchical algorithm supports a network dataset composed of up to three levels of hierarchy. When creating a new network dataset, users must specify how they would like to create the hierarchy levels of their edge sources. The following two steps are fundamental to create a network dataset with a hierarchical classification.

- First, a new type of network attribute, called the hierarchy attribute, is used to assign a rank, an order of importance such as the road classification, to each network edge element. This rank is represented by an integer value. The lowest integer value represents the most preferred network element.

- Then, the hierarchy ranges indicate how to group the different hierarchy values into three levels of hierarchy. The build algorithm will cluster the hierarchical levels based on those hierarchy ranges. The default ranges assign the hierarchy values 1, 2, and 3, respectively, to hierarchy level 1 (primary roads), hierarchy level 2 (secondary roads), and hierarchy level 3 (local roads). Users can specify custom ranges according to their data.

The image on the left side in figure 6 shows an example of hierarchy modeling using the default hierarchy ranges. The second image, on the right side, shows a case where the hierarchy attribute has five values (1, 2, 3, 4, and 5) that represent five road classes (interstates, highways, major arterial roads, local collectors, and feeders) and are regrouped into three levels of hierarchy.

Figure 6
Default Hierarchy Ranges (left) and Custom Hierarchy Ranges (right) Set When Creating the Network Dataset in ArcCatalog



Notes

- The default ranges specified when the network dataset was created can be modified at analysis time to adjust the hierarchy ranges appropriately. However, changing the hierarchy ranges before a solve operation will not change the initial spatial clustering created by the network build algorithm.

- When using commercial street data, a road classification field may help to define the different network groups to compute logical and efficient routes. This field can be used to assign values to a hierarchy attribute (see Appendix B, Examples of Hierarchy Modeling).

- If a network dataset has multiple source feature classes, the hierarchy attribute evaluator must be carefully assigned. For example, for a network dataset composed of three street feature classes (primary roads, secondary roads, and local roads), the hierarchy evaluator will be constant for each source and set respectively to 1, 2, and 3.

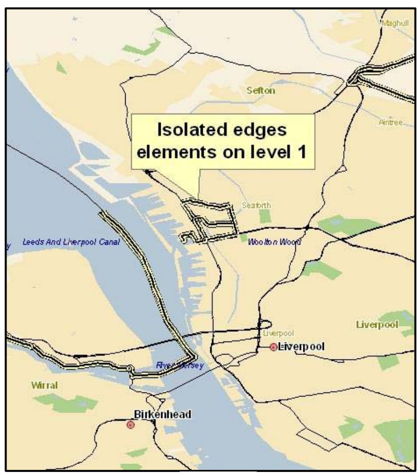
- Modeling hierarchical levels when multiple network modes are included in one network dataset (e.g., streets for car and pedestrian navigation, metro lines and train lines for public transit navigation) can be a challenging task that involves planning. For this task, users would have to analyze which travel modes will be included, or excluded, at each level of hierarchy. Furthermore, users would have to analyze whether transferring between different network modes and through the different levels of hierarchy is permissible. In other words, an edge representing a transfer from one mode (e.g., metro line) to another mode (e.g., train line) should be coded with the appropriate hierarchy value to yield expected routing results.

Modeling Considerations

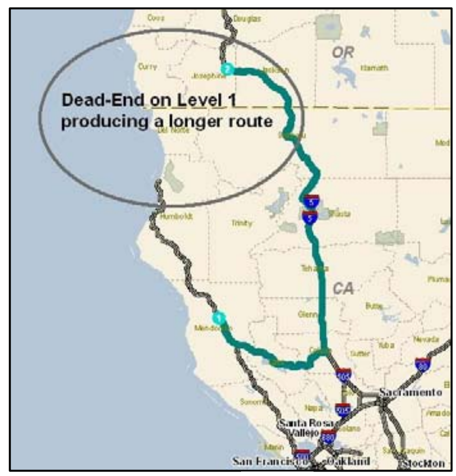
The hierarchical best route algorithm implemented in Network Analyst requires properly built hierarchical levels to produce reasonable results. Otherwise, it may produce substantially longer routes, taking significant computing time, or in the worst-case scenario, it may fail to find a solution. It is important to analyze the source data prior to building the network dataset. Following are some practical guidelines on building a network dataset with hierarchy listed by order of importance:

- Check if the edge elements in the network sources are properly connected. For example, there should be no islands (isolated edge elements) on hierarchy level 1. Similarly, dead ends on levels 1 and 2 affect route quality. It is also important to verify that there is ascendant hierarchy compatibility. For example, if a ramp access between a primary road and a secondary road is coded as a local road, it will not be possible to get on the primary road from this secondary road.

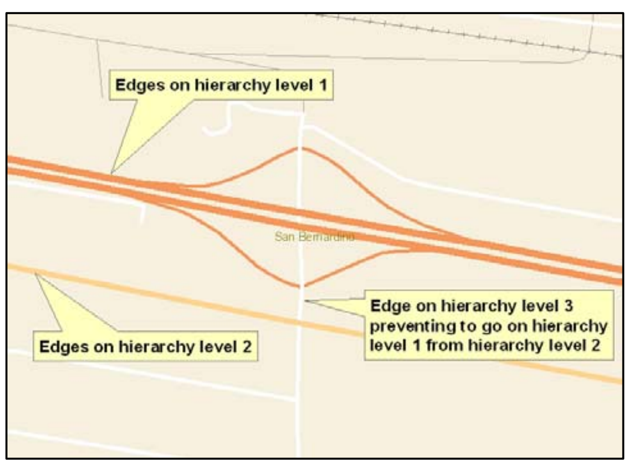
Figure 7
Examples of Disconnected Levels of Hierarchy Due to Isolated Edges (a), Dead End (b), and No Ascendant Connectivity between Hierarchy 2 and 1 (c)



a.



b.



c.

- Create a hierarchy attribute and assign to it the appropriate values. The evaluator must be created carefully; it assigns the hierarchy attribute value to each network edge element. For example, a hierarchy attribute can be assigned using a field, set to a constant value, or set using a VBScript evaluator.

- Have a good ratio of edges belonging to the different hierarchical levels. A pyramidal approach is recommended; each higher level contains a fewer number of edge elements than the previous level (approximately 10–20% of the previous level). However, the highest level of hierarchy should have enough edges to provide good route quality. For example, the nationwide U.S. road network should have approximately 500,000 edge elements in hierarchy level 1 when using commercial data.

- Define which cost and restriction attributes can be used during a solve operation with the hierarchy algorithm. Hierarchical networks and cost attributes are inherently tied together, meaning that certain cost attributes may not be used during analysis.

- Understand that disconnected network elements can occur at solve time when using a specific cost attribute with negative values as impedance and/or restriction attributes. Furthermore, disconnected network elements may be due to custom hierarchy ranges set by users.

StreetMap Network Considerations

StreetMap networks can be consumed by Network Analyst similar to other network dataset formats (personal geodatabase, enterprise geodatabase, and shapefile). This means that StreetMap datasets provided with ArcGIS StreetMap can also be used for routing with Network Analyst. Users cannot build StreetMap networks within the ArcGIS Network Analyst framework. In addition, you can contact ESRI, ESRI distributors, or commercial data vendors to purchase other street datasets in StreetMap file format. Visit www.esri.com/data/index.html.

StreetMap networks, based on ESRI's Spatial Data Compression (SDC) release 2.0, may either have single or multiple SDC levels including routing topology. If a StreetMap network has only a single SDC level, then the hierarchical algorithm cannot be used when solving a Route, Closest Facility, or OD Cost Matrix problem. If a StreetMap network has multiple SDC levels, then the hierarchical algorithm can be used for routing analyses on small or large datasets with good performance.

Notes

- In StreetMap networks, the multilevel classification is opposite of that for "standard" network datasets. The first SDC level is the basic level containing all the street segments. When accessing a StreetMap network from Network Analyst, the default ranges are automatically adjusted to reverse the hierarchy classification.

- Very large StreetMap networks (e.g., North American road network, Western European road network) are often modeled with four SDC levels of hierarchy. AsNetwork Analyst can handle only three levels, the hierarchy ranges are assigned as follows:

SDC Level	Hierarchy Group
1st SDC level	Local roads
2nd SDC level	Secondary roads
3rd SDC level	Secondary roads
4th SDC level	Primary roads

Using Hierarchy in Network Analysis

ArcGIS offers a complete and flexible framework in which network analyses can be performed in ArcMap, using a Geoprocessing tool, in a custom application developed with ArcGIS Engine, or via a Web application developed using ArcGIS Server. Independently of how the network analysis is performed in ArcGIS, it is important to understand how to efficiently use the hierarchy settings to get accurate routes with optimal performance. The hierarchy settings are available in the Network Analyst Route, Closest Facility, and OD Cost Matrix solvers.

This section lists the best route algorithm parameters exposed to users and discusses the importance of those parameters when performing an analysis. It also explains how to assess the accuracy of computed routes and measure the performance. It concludes with tips on diagnosing and fixing the causes of unexpected solver behavior.

Best Route Algorithm Parameters

Table 1 lists the common parameters for performing analyses in Network Analyst with the exact or hierarchical best route algorithms. This table also gives the definition of each parameter, its default value, and whether it is required in an analysis.

The minimum requirement to perform an analysis is to specify the impedance to minimize. The restriction attributes and the accumulate attributes are optional. The U-turn policy is automatically set to its default value.

Table 1
Common Parameters in ArcGIS Network Analyst Functions

Parameter Name	Definition	Default Values	Required
Impedance	Impedance minimized while determining the route. Any cost attribute can be chosen as impedance.	N/A	Yes
Restriction Attributes	List of restriction attributes used for limiting traversal through a network dataset (e.g., "one way street," "no trucks allowed," "restricted turns," and "buses only").	N/A	No
Accumulate Attributes	List of cost attributes accumulated once the route is computed. For example, if the cost attribute DriveTime is selected as impedance and another cost attribute Length is accumulated, then route will return the Total_DriveTime and the Total_Length.	N/A	No
U-Turn Policy	While calculating a route, the U-turn policy can be set to everywhere, nowhere, or only at dead ends (also known as cul-de-sacs).	Everywhere	Yes

Table 2 lists the parameters specifically used to invoke the hierarchical algorithm. These parameters are only available in the Route, Closest Facility, or OD Cost Matrix solvers.

The minimum requirement to compute a route using the hierarchical algorithm is to set the parameter Use Hierarchy. The other parameters, Hierarchy Name, Hierarchy Ranges, and Number of Transitions to Hierarchy, are either optional or automatically set to their default values.

Table 2
Parameters Used to Invoke the Hierarchical Algorithm in ArcGIS Network Analyst

Parameter Name	**Definition**	**Default Values**
Use Hierarchy	Used to invoke the hierarchical best route algorithm during analysis. This option is enabled only if the network dataset has a hierarchy attribute.	False in ArcObjects™ True in ArcMap and Geoprocessing tools
Hierarchy Name	Specify the hierarchy attribute name.	N/A
Hierarchy Ranges	Ranges used to reclassify network edge elements before finding a hierarchical route. The hierarchy ranges are defined by two parameters specifying the hierarchy attribute values belonging respectively to hierarchy levels 1 and 2. The hierarchy ranges are specified when the network dataset is created, and these ranges become the default values during analysis. If the user does not set the hierarchy ranges when building a network dataset, these following values are used to provide an initial hierarchy clustering: ■ All edges with a hierarchy attribute value lower or equal to 1 are in level 1 (primary roads). ■ All edges with a hierarchy attribute value of 2 are in level 2 (secondary roads). ■ The rest of the edges fall in level 3 (local roads). In ArcCatalog™ and ArcMap, the hierarchy ranges can be set in a dialog box. In ArcObjects, developers can set the hierarchy ranges using the property MaxValueForHierarchy(1) for hierarchy level 1 and the property MaxValueForHierarchy(2) for hierarchy level 2.	Defined when the network dataset is built

Parameter Name	Definition	Default Values
Number of Transitions to Hierarchy	Maximum number of entry (or exit) points to reach (or exit) a given level of hierarchy from (or to) a lower level of hierarchy. There are two parameters, available only through ArcObjects, defining ■ The maximum number of points to reach (or exit) level 1 from (to) levels 2 or 3, NumTransitionToHierarchy(1). ■ The maximum number of points to reach (or exit) level 2 from (to) level 3, NumTransitionToHierarchy(2).	NumTransition ToHierarchy(1) = 9 NumTransition ToHierarchy(2) = 6

Importance of Parameters

The importance of the following three groups of parameters, defined in table 1 and table 2, when computing a route is discussed below. The three groups of parameters are

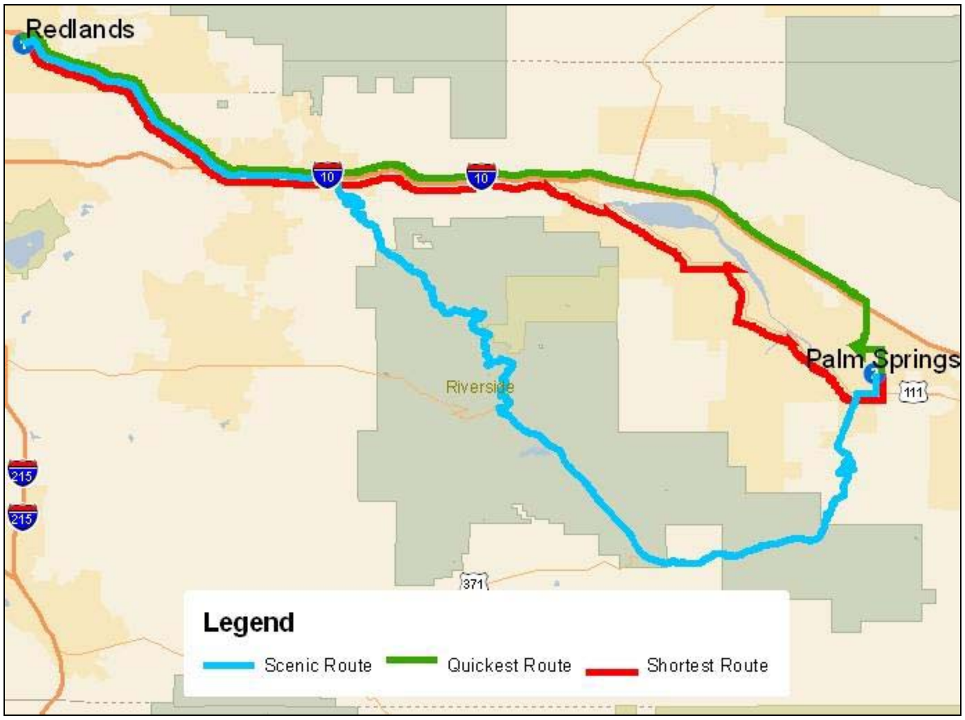
- The impedance to minimize
- The restriction attributes to apply, the cost attributes to accumulate, and the U-turn policy
- The hierarchy options (use hierarchy, hierarchy ranges, and number of transitions to hierarchy)

Impedance

The most important parameter is the impedance to minimize. The computed route shape, its total cost, and the related driving directions depend on the cost attribute selected as impedance. Figure 8 shows an example of three different routes between the same set of stops computed with three different cost attributes, namely DriveTime, Distance, and Scenic (a low-cost value on edges indicates a high degree of interest for tourist). These three routes represent the quickest route, the shortest route, and the most scenic route offering panoramic view. They are each optimal for the chosen impedance attribute. They can be suboptimal if the hierarchy algorithm is invoked.

J-9458

Figure 8
Example of Three Different Routes (Quickest, Shortest, and Most Scenic)



Quickest (Drive Time = 1 hour, 7 minutes; Distance = 94.7 kilometers), Shortest (Drive Time = 1 hour, 12 minutes; Distance = 92.5 kilometers), and the Most Scenic Routes (Drive Time = 2 hours, 6 minutes; Distance = 143 kilometers) between Redlands and Palm Springs, California

Furthermore, users should be aware that hierarchical levels and cost attributes are inherently tied together. For example, consider a network dataset that has been built with a time-based cost attribute, called DriveTime, and a hierarchy attribute, called FavorMajorRoads. The DriveTime attribute is computed with a formula involving edge length, road network classification, and speed limits, while FavorMajorRoads is based on road network classification, which itself is related to the speed limits. There is a direct relationship between the cost attribute and the hierarchy attribute in that a shortest route on the DriveTime cost attribute will use the fastest edge elements in the network, whether the hierarchy options are used or not.

Conversely, it can be irrelevant to use some cost attributes when using the hierarchy options. For example, it does not make sense to use the hierarchy option favoring primary roads while searching for a pedestrian route in heavily urbanized areas.

Restriction Attributes, Accumulate Attributes, and U-Turn Policy

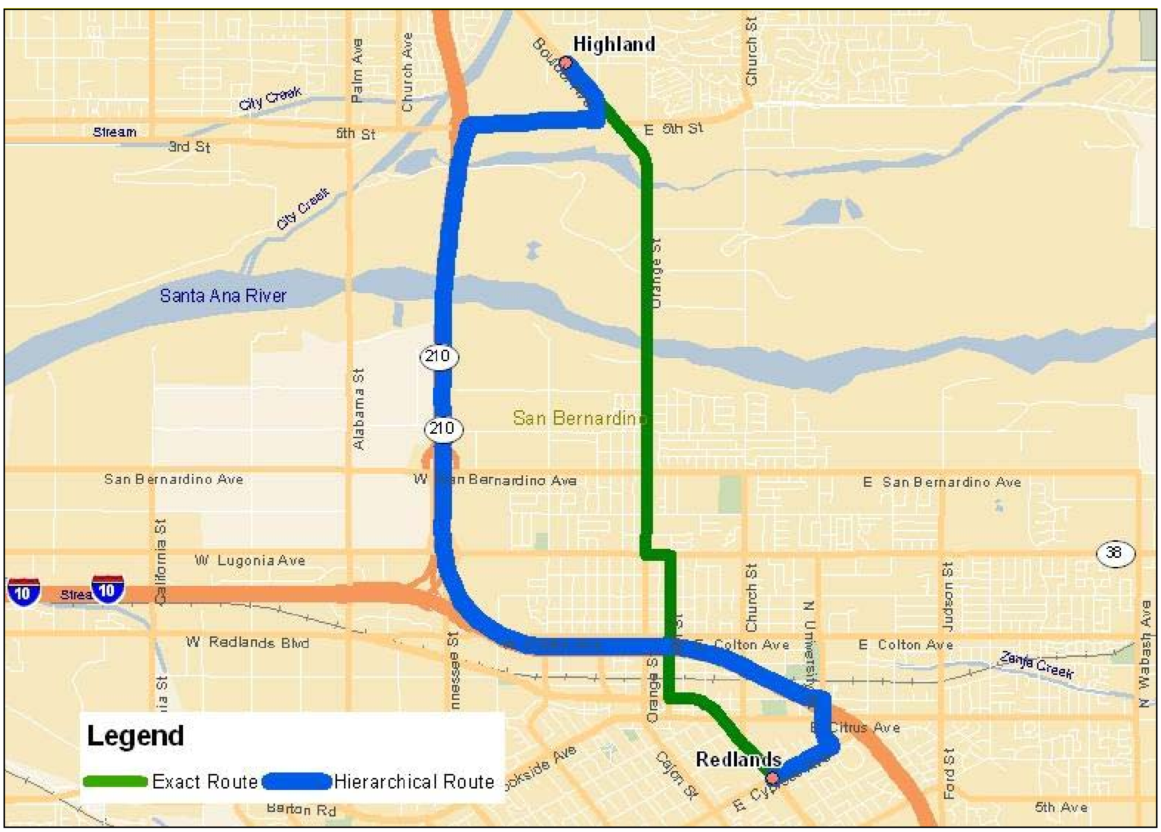
To get an accurate route while routing vehicles, it is important to apply the appropriate restrictions such as one-way and turn restrictions. Also, the U-turn policy may impact the route solution. The reader can refer to ArcGIS Network Analyst Online Help, 2005, for more details about the U-turn policy.

Accumulating cost attributes does not have any impact on how the route is computed.

Hierarchy Options

Using the hierarchy option during analysis (use hierarchy parameter checked on in ArcMap) will impact both the solution quality and computation time. The route can be suboptimal (i.e., longer in terms of impedance, but may be more realistic). Figure 9 shows the quickest route computed both with the exact route solver and the hierarchical route solvers. Furthermore, using the hierarchical algorithm will significantly decrease the computation time. For example, the hierarchical solver is one order of magnitude faster than the exact solver when finding a route across a Southern California network with 1.3 million edges.

Figure 9
Example of Exact Route versus Hierarchical Route



Exact Quickest Route (Drive Time = 8 minutes, 30 seconds) versus Hierarchical Quickest Route (Drive Time = 10 minutes, 15 seconds) from Redlands to Highland, California

Both the hierarchy ranges and the number of transitions between hierarchy levels play an important role in how the hierarchical algorithm behaves. These parameters directly impact the route quality and the computation time.

The hierarchy ranges indicate how to group the different levels of hierarchies based on the hierarchy attributes values.

As explained in the subsection Modeling Considerations, the hierarchical decomposition of the network must respect a pyramidal approach, with each higher hierarchy level having fewer edges than its previous level, to obtain lower computational time. However, a minimum set of edge elements is required in each level to get good solution quality. On the other hand, too many edge elements on the highest level of hierarchy may increase the computation time. The trade-off between speed and quality may depend on various factors such as the user's goals, data quality, and data extent. Several attempts may be needed to determine good hierarchical ranges for a particular solution goal. For example, the three following steps can be repeated to get both good performance and accurate routes:

1. Build the network dataset with one set of hierarchy range values.

2. Compute a set of well-defined routes.

3. Analyze the solution quality and the performance of the computed routes (refer to Assess Solution Quality and Performance).

Once the hierarchy ranges have been chosen and the network dataset created, the ranges become the default setting when a route is computed. It is possible to set custom modified hierarchy range values before computing a route and without rebuilding the network dataset. For example, suppose the road network of the United States has been built with only the interstate roads as primary roads to get fast computation time for a long-distance routing application. However, if the origin and destination stops are in the same local area, a user can adjust the hierarchy ranges so the interstate and local major roads are now part of the primary roads. That will perhaps return a more accurate route while increasing the computation time because more edges are included in hierarchy level 1 and explored during the search.

Figure 10
Example of Hierarchical Route Using Default and Custom Hierarchy Ranges

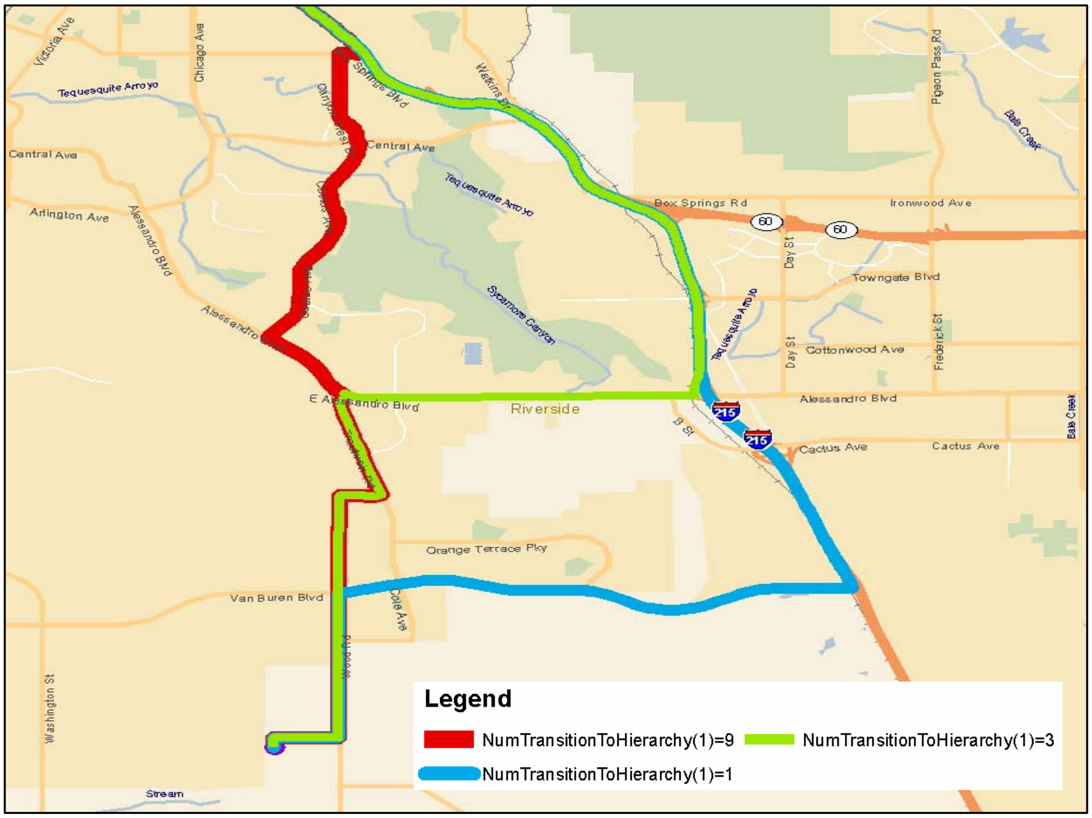
The number of transitions to hierarchy limits the traversal search on each level of hierarchy as explained in the subsection Overview of the Hierarchical Algorithm. The higher the values are for hierarchy levels 1 and 2, the better the accuracy of the computed route. On the other hand, low values may affect the accuracy of the route but speed up the solve operation.

The values for the number of transitions for hierarchy of levels 1 and 2 are set to their default values (see table 2), when creating a Network Analyst layer using the Network Analyst user interface, programmatically with ArcObjects, or using Geoprocessing tools. The default values have been defined to cover the most typical use cases based on extensive route computations on various networks in different data formats (e.g., NAVTEQ, Tele Atlas, and proprietary data), with different sizes (from 100,000 edges to 40 million edges), and with different areas (North America and Europe).

The default number of transitions to hierarchy level 1 is greater than the default for hierarchy level 2, respectively 9 and 6. Generally for road networks, hierarchy level 1 is composed of an important number of one-way roads (e.g., interstates, motorways, freeways, local major roads). This means that the hierarchical algorithm requires a larger number of entries onto the highest level to take into account one-way restrictions and to return a more accurate result.

The default values cannot be changed by users with the Network Analyst user interface or Geoprocessing tools. They can be modified only programmatically with ArcObjects for specific needs (see table 2). If an application requires changing the default values, it may be important to compute a set of well-defined routes and analyze both accuracy and performance of the computed routes for different settings of the number of transitions of hierarchy values.

Figure 11
Example of Routes Computed with Different Number of Transition to Hierarchy Values



Example of different routes to reach a destination from an origin computed with three different values of the number of transition to hierarchy 1. The route computed with the default value (NumTransitionToHierarchy(1)=9) is the fastest 42 mn, 39 s versus 45 mn, 15 s and 45 mn, 25 s, respectively, for NumTransitionToHierarchy (1)=3 and NumTransitionToHierarchy (1)=1. In this case, higher NumTransitionToHierarchy(1) than 9 does not return a better solution.

Assess Solution Quality and Performance

The validation process is an important task when developing a routing application. The validation process may include assessing the solution quality and performance. Assessing solution quality may depend on the type of application developed such as how close the computed solution is to optimal and how feasible and realistic the computed solution is. Assessing performance can involve measuring the computation time and evaluating the resources (physical memory size, CPU speed, hard drive size and speed, etc.) needed to configure the deployment machine.

Following are guidelines providing practical information to assess the solution quality and the computation time of the hierarchical best route algorithm. These recommendations can be performed using a well-known set of stops and compared with expected results, using exotic cases (solve routes on areas where there is a sparse primary road network), and may be intensively repeated (e.g., for server-based applications). Furthermore, users may relate these recommendations to ones provided in the subsection Modeling Considerations.

- **Visual inspection of the routes in ArcMap.** The route must minimize the cost and honor restrictions if turned on such as one-way restrictions and prohibited maneuvers. Also, it is important to verify that the routes go onto and off of the highest hierarchical network properly.

- **Compare exact versus hierarchical routes.** It is important to measure both the error in terms of the total cost difference and the computation time difference between the two solves for the same origin and destination. That may help to determine if the primary roads network and the secondary roads network have a good ratio of edges elements versus the local road networks (respect a pyramidal approach).

- **Compute routes in high-density network areas (urban areas) and in low-density areas (rural areas).** This may help to determine if there is a good spatial coverage of edge elements in the primary roads network.

- **Change the default hierarchy range** values and measure the impact on the solution quality and the solve time versus the exact route solver and the hierarchical route using default ranges. This may help to evaluate if the network dataset needs to be rebuilt with new default hierarchy ranges.

- **Shut down street portions, entire streets, major intersections, or areas with barrier network locations and solve routes.** Here the goal is to check if good alternative routes are computed when barriers are involved.

Diagnostic Problems

The following provides additional guidelines to diagnose the causes of unexpected behavior that users may encounter when using the new hierarchical best route algorithm in Network Analyst. Unexpected behavior includes

- An inaccurate route is computed.
- The application takes too much time to compute the solution.
- In a worst-case scenario, no route is found.

The causes of such problems can vary: data modeling issues, software limitations, or the route is simply infeasible. For example, one may think that the result of a route analysis using hierarchy is too long in terms of total cost minimized. That may be due to incorrect hierarchical modeling in the network dataset or due to the hierarchal best route algorithm itself, which is heuristic and does not guarantee optimality.

The list below is intended to help users analyze the causes of an unexpected route between two stops computed by the hierarchical best route algorithm and to find a solution to fix the underlying problem.

- Verify that the appropriate cost attributed is used as impedance.

- Verify that the restriction attributes are correctly modeled (use the Network Identify tool in ArcMap), are used when performing the analysis (checked on), and are honored by the route.
- Verify that there is a hierarchy attribute in the network dataset and that the Use Hierarchy option is used during analysis.
- Verify that hierarchy level 1 (primary roads) is not disconnected. Display hierarchy level 1 using the feature class source(s). Use the Network Identify tool in ArcMap; it allows you to verify if the hierarchy attribute values are correctly assigned.
- Verify that hierarchical levels 1 and 2 have neither too many nor too few edge elements (respect a pyramidal approach).
- Verify that the exact route solver finds an optimal solution with the same stops.
- If solving with barriers, verify that there is a feasible route without barriers.
- Solve with the same stops with custom hierarchy ranges. For example, if using StreetMap data with four SDC levels of hierarchy, set the range for primary roads to be up to 2.
- Rebuild the network dataset using a different source data format (shapefile, personal, or enterprise geodatabase), then resolve the same problem.
- Solve the same problem using data located on the same machine as the application.

Summary

The goal of this paper is to provide information to desktop users or developers building stand-alone or server applications to effectively use the hierarchical route solver implemented in ArcGIS Network Analyst.

The paper began with a brief comparison of exact and hierarchical route algorithms. It next presented the underlying algorithmic concepts allowing users to solve a route between an origin and a destination or to compute all routes between multiple origins and multiple destinations, the n by m route problems, namely Closest Facility or Origin-Destination Cost Matrix in ArcGIS Network Analyst. This section explained how the hierarchical algorithm handles barriers to avoid restricted network elements and also turn restrictions that prohibit maneuvers between a set of network edge elements.

Then, this paper discussed how to properly build a network dataset including multiple levels of hierarchy. First, this section explained how to create a hierarchy attribute and to specify the default hierarchy ranges when the network dataset is built. Second, this section presented practical guidelines to analyze the data prior to creating the hierarchy levels (find the disconnected network elements or determine the number of edge elements per hierarchy level). Finally, this section explained how StreetMap network datasets are modeled.

In the last section, this paper explained how to use the hierarchical route solver to get accurate routes and optimal performances in ArcGIS Network Analyst. This section described the list of parameters of the hierarchical route solver exposed to users and discussed the importance of those parameters when performing an analysis. It also explained how to assess the accuracy of computed routes and the performance in terms of computation time and memory consumption and concluded with tips on diagnosing and fixing the causes of unexpected solver behavior.

Appendix A: Details of the Hierarchical Route Algorithm

This appendix details how the hierarchical solver computes the best route between an origin and a destination. The technique consists of

- Simultaneously growing two traversal trees, also called shortest path trees, based on the Dijkstra method—one forward traversal tree from the origin and one backward traversal tree from the destination—until K_2 entries on hierarchy level 2 or K_1 entries on hierarchy level 1 are found for each tree.

- Applying a hierarchy filter to restrict the search to edges in hierarchy levels 1 or 2. In this phase, the forward and backward traversal trees are grown until K_1 entries on hierarchy level 1 are found for each tree.

- Applying a hierarchy filter to restrict the search to edges in hierarchy level 1. In this phase, only the forward traversal tree is grown until the best route is found.

During the search, a best route from the origin to the destination is maintained. Each time the forward traversal tree meets the backward traversal tree, the best route is updated, and this phase stops when you have determined that the best route cannot be improved. This process includes all feasible routes found in phases 1 and 2.

Notes

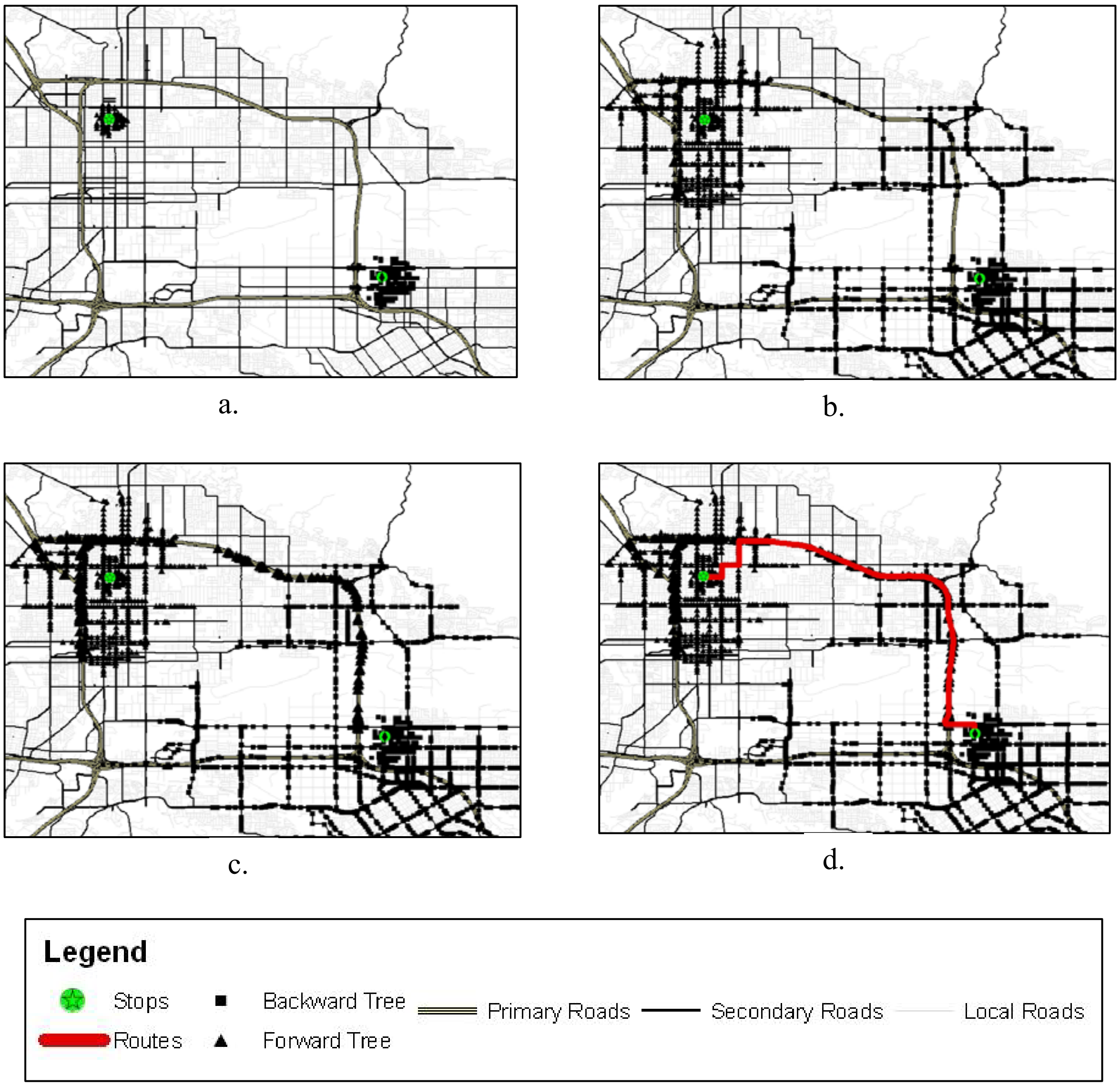
- The first step of the hierarchical algorithm is equivalent to a symmetrical or bi-directional Dijkstra algorithm introduced by Pohl[1]. This algorithm computes an optimal route without exploring the entire network. So, if the origin and destination are close enough, meaning that steps 2 and 3 are not executed, an optimal solution may be found without executing phases 2 and 3.

- Higher values of K_1 and K_2 produce more accurate routes. In ArcObjects, K_1 and K_2 are exposed as properties respectively named NumTransitionToHierarchy(1) and NumTransitionToHierarchy(2).

- This strategy assumes that the highest order of hierarchy is connected. As long as level 1 is fully connected, every edge on level 2 is connected to at least one edge on level 1, and every edge on level 3 is connected to at least one edge on level 2, the hierarchical algorithm will always find a route. On the other hand, if level 1 is disconnected, the hierarchical algorithm will not descend to the lower levels when a dead end is encountered on level 1. In a worst-case scenario, it will not find a route.

[1] Pohl, I. "Bi-directional Search," in *Machine Intelligence*, Vol. 6, eds. Meltzer and Michie (Edinburgh University Press, 1971), 127–140.

Figure 12
Example of Hierarchical Bidirectional Search



A, b, and c show, respectively, the forward and backward traversal trees grown during phases 1, 2, and 3. The last figure (d) shows the assembled best route.

Appendix B: Examples of Hierarchy Modeling

This appendix provides guidelines on how to build a network dataset with hierarchy using four different data formats: ESRI ArcLogistics Route 3 map data format[1], Dynamap/Transportation versions 6.x and 7.0 by Tele Atlas[2, 3], MultiNet version 4.2 by Tele Atlas[4], and NavStreets version 3.3 by Navigation Technologies[5].

The table below indicates which source layer and fields can be used to create the three levels of hierarchy per data format.

Data Format	**Layer to Use**	**Field to Use as Evaluator**	**Hierarchy Settings**
ESRI ArcLogistics Route 3	Allst	Code for classifying streets (Disp_Code)	Level 1: Disp_Code = 10 or 20 Level 2: Disp_Code = 30 Level 3: Disp_Code >= 40
Dynamap/Transportation 6.x	Street	Arterial Classification Code (ACC) Feature Class Code (FCC)	Level 1: ACC = 1 or 2 Level 2: (ACC = 3 or FCC < A40 or FCC = A60 or FCC = A63 or FCC = A68 or FCC = A69 or FCC = A45 4) and ACC <> 1 and ACC <> 2 Level 3: All edges, excluding all edges set in levels 1 and 2
Dynamap/Transportation 7.0	Street	ACC	Level 1: ACC = 1 or 2 Level 2: ACC = 3 or 4 Level 3: ACC = 5

1 *ArcLogistics Route 3 Map Data Specification,* ESRI Technical Paper, 2001.
2 Dynamap/Transportation *Product User Guide,* Version 6.x, Tele Atlas, 2004.
3 Dynamap/Transportation *Product User Guide,* Version 7.0, Tele Atlas, 2005.
4 MultiNet, *User Guide,* Tele Atlas, 2004.
5 *NavStreets Product Guide,* Version 3.3, Navigation Technologies, 2005.

Data Format	**Layer to Use**	**Field to Use as Evaluator**	**Hierarchy Settings**
MultiNet 4.2	Network	Functional Road Class (FRC) for North American and European networks	Level 1: FRC = 0, 1 or 2 Level 2: FRC = 3 or 4 Level 3: FRC >= 5
		or Network Classification (Net2Class) for European networks only	Level 1: Net2Class = 1 or 2 Level 2: Net2Class = 3 or 4 Level 3: Net2Class = 5
NavStreets 3.3	Streets	Functional Class (Func_Class)	Level 1: Func_Class = 1 or 2 Level 2: Func_Class = 3 or 4 Level 3: Func_Class = 5

ESRI

For more than 35 years ESRI has been helping people manage and analyze geographic information. ESRI offers a framework for implementing GIS technology in any organization with a seamless link from personal GIS on the desktop to enterprise-wide GIS client/server and data management systems. ESRI GIS solutions are flexible and can be customized to meet the needs of our users. ESRI is a full-service GIS company, ready to help you begin, grow, and build success with GIS.

Corporate

ESRI
380 New York Street
Redlands, California
92373-8100, USA
Telephone: 909-793-2853
Fax: 909-793-5953

For more information
on ESRI, call

1-800-447-9778

(1-800-GIS-XPRT)

or contact an ESRI value-added reseller near you.

Send e-mail inquiries to

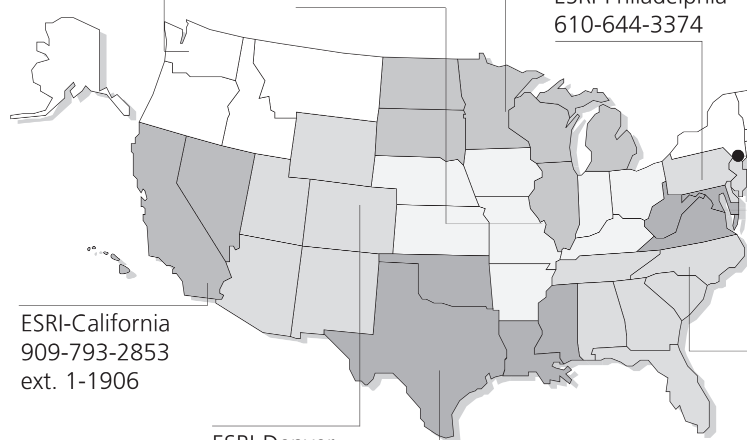
info@esri.com

Visit ESRI's Web page at

www.esri.com

Outside the United States, contact your local ESRI distributor. For the number of your distributor, call ESRI at 909-793-2853, ext. 1-1235, or visit our Web site at
www.esri.com/international

Regional Offices



International Offices

Australia
www.esriaustralia.com.au

Belgium/Luxembourg
www.esribelux.com

Bulgaria
www.esribulgaria.com

Canada
www.esricanada.com

China (Beijing)
www.esrichina-bj.cn

China (Hong Kong)
www.esrichina-hk.com

Finland
www.esri-finland.com

France
www.esrifrance.fr

Germany/Switzerland
www.esri-germany.de
www.esri-suisse.ch

Hungary
www.esrihu.hu

India
www.esriindia.com

Indonesia/Malaysia
62-21-527-1023
603-7874-9930

Italy
www.esriitalia.it

Japan
www.esrij.com

Korea
www.esrikr.co.kr

Netherlands
www.esrinl.com

Poland
www.esripolska.com.pl

Portugal
www.esri-portugal.pt

Romania
www.esriro.ro

Singapore
www.esrisa.com

Spain
www.esri-es.com

Sweden
www.esri-sweden.com

Thailand
www.esrith.com

United Kingdom
www.esriuk.com

Venezuela
www.esriven.com

Place ESRI business partner or distributor address here.

GSA Federal Supply Schedule
No. GS-35F-5086H

Printed in USA